Image Correlation Quality Control enabled by Cluster Computing

Gerhard Klimeck, Gary Yagi, Robert Deen, Myche McAuley, Eric DeJong, Fabiano Oyafuso

Stereo image correlation is an essential ingredient to mars missions for mapping and ranging purposes. Parallelization of available correlation software reduced the required processing time from about 60 to 90 minutes to 2-3 minutes per image pair. A quality control of the correlation has been previously too time intensive, but is now enabled by the fast parallel execution. The stereo image correlation software attempts to deliver for every pixel (x,y) in the reference image (left or right) a corresponding pixel coordinate (x',y') in the corresponding pair (right or left). This is performed in a parallel algorithm that subdivides the reference image onto different CPUs. The important thing to realize is that the quality of the mapping from left image pixels to right image pixels may not be perfect due to several reasons. Two of the major reasons are: 1) the left eye might see objects that the right eye does not, and vice versa (parallax), and 2) the image data might be noisy. The correlation / mapping output delivered by the correlation software may therefore be faulty. The quality of the correlation is assessed by checking a left->right correlation by a right->left correlation. It is based on establishing TWO correlation maps left->right and right->left, therefore doubling the overall workload. The originally desired mapping relates the point (x,y) to the point (x',y'). The corresponding mapping from (x',y') back to the left will result in the pixel (x",y"). If (x,y) and (x",y") are identical we consider this mapping a perfect correlation. The very hard criterion of a perfect match can be relaxed with the introduction of an error window, which accounts for integer interpolation round-off and a small amount of noise. The procedure can be repeated until all unconnected point-to-point connections are eliminated, resulting in verified correlation data.

JPL

ACT

Image Correlation Quality Control enabled by Cluster Computing

Gerhard Klimeck,
Gary Yagi, Robert Deen, Myche McAuley, Eric DeJong, Fabiano Oyafuso
Jet Propulsion Laboratory, California Institute of Technology

Applied Cluster Computing Technologies Group (381)
Multimission Image Processing Lab (382)

This research was carried out by at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

Work sponsored by the TMOD Technology Program under the Beowulf Application and Networking Environment Task And the Computational Technologies Program under ESTO.

JPL

ACT

Mars Image Processing on Clusters

Need

- **Mosaics and terrain data from sequences of images.**
- **Rapid turnaround for the rover navigation team in operations.**

Problem

- **Current technology requires about 90 minutes per mosaic and 70 minutes per correlation pair.**
- **Requirements are 30 minutes for 2.5x larger datasets.**

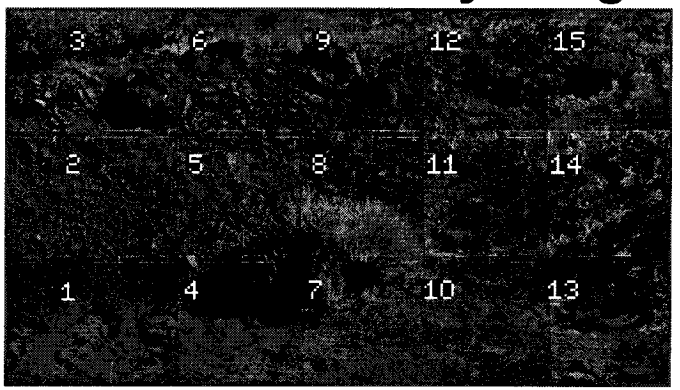
Approach

- **Parallelize on a commodity cluster.**

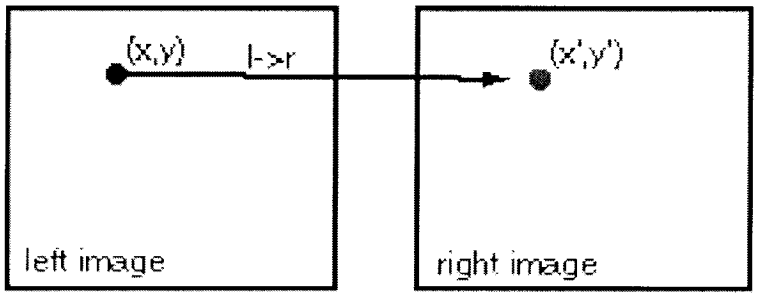
Outline of Presentation

- **Reduction of mosaic times.**
- **Reduction of correlation times.**
- **Enable correlation quality control.**
- **Increase correlation algorithm stability.**

Mosaics from many images

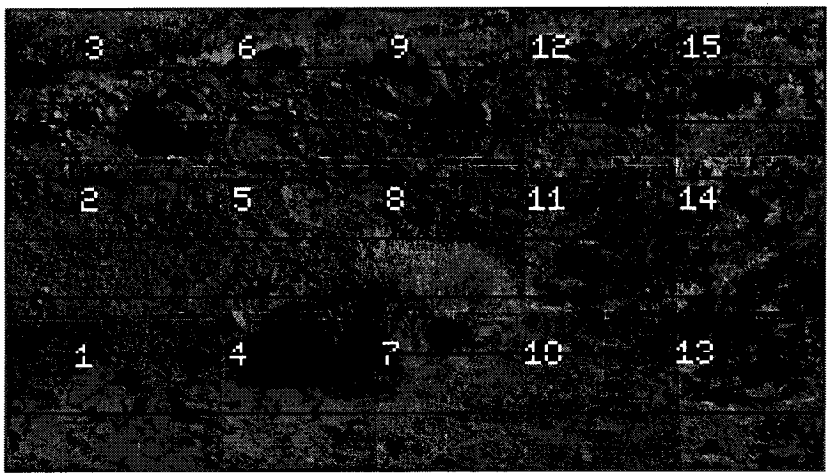


Correlation of Stereo Pairs



JPL

Mars Mosaic Images

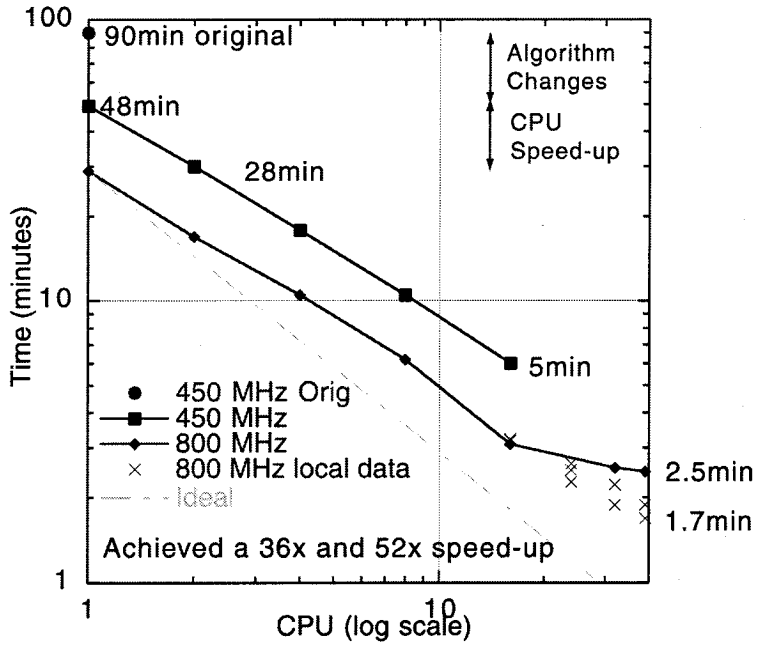


Problem:

- Need to stitch together one single image from hundreds of individual images
- Software exists, but but takes 90minutes to assemble one image from 120 images.

Approach:

- Optimize existing algorithm.
- Parallelize existing software
  - Subdivide image into slices 1 slice/CPU



Results:

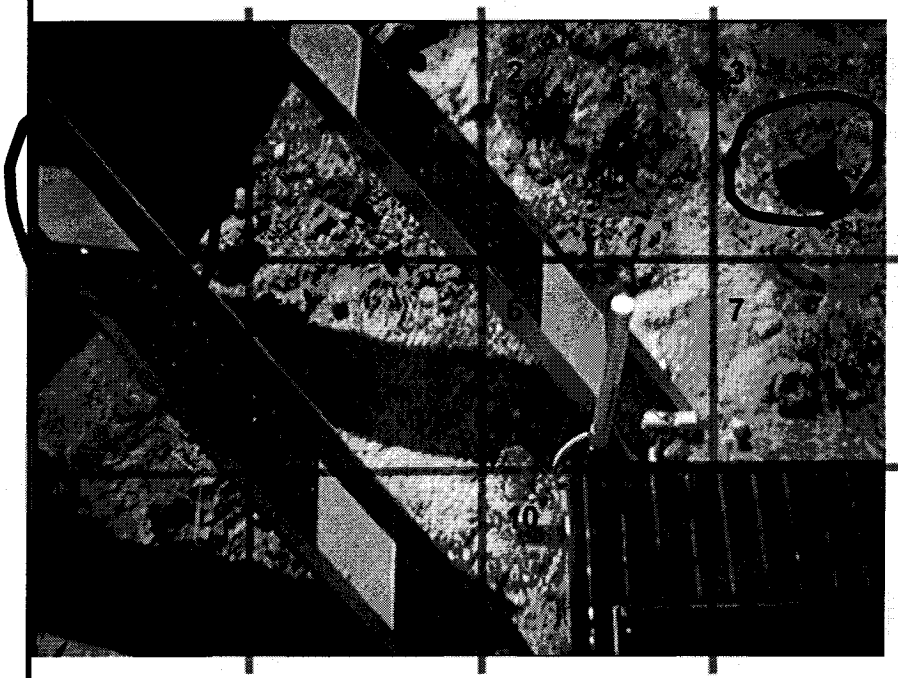
- Starting point 450MHz: 90 minutes
- Algorithm Changes -
  Storage of all images: 48 minutes
- 16 CPUs / 450MHz: 5 minutes
- 40 CPUs / 800MHz: 2.5 minutes
- with local data:
  40 CPUs / 800MHz: 1.7 minutes /

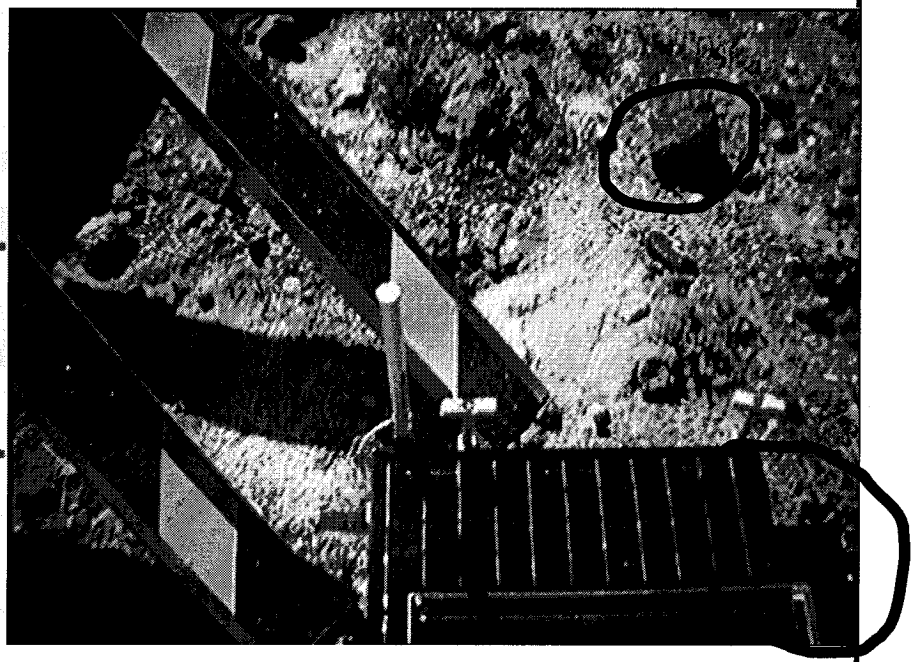
Impact:

- Near real-time processing of mosaics.
- Software part of the processing pipeline

JPL

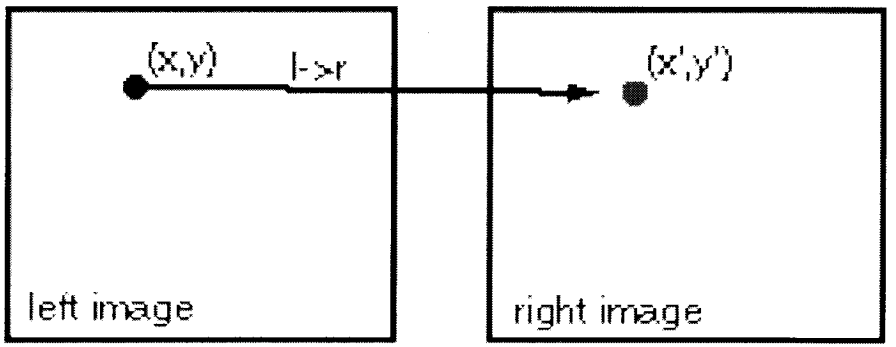
Left & Right Stereo Image Correlation





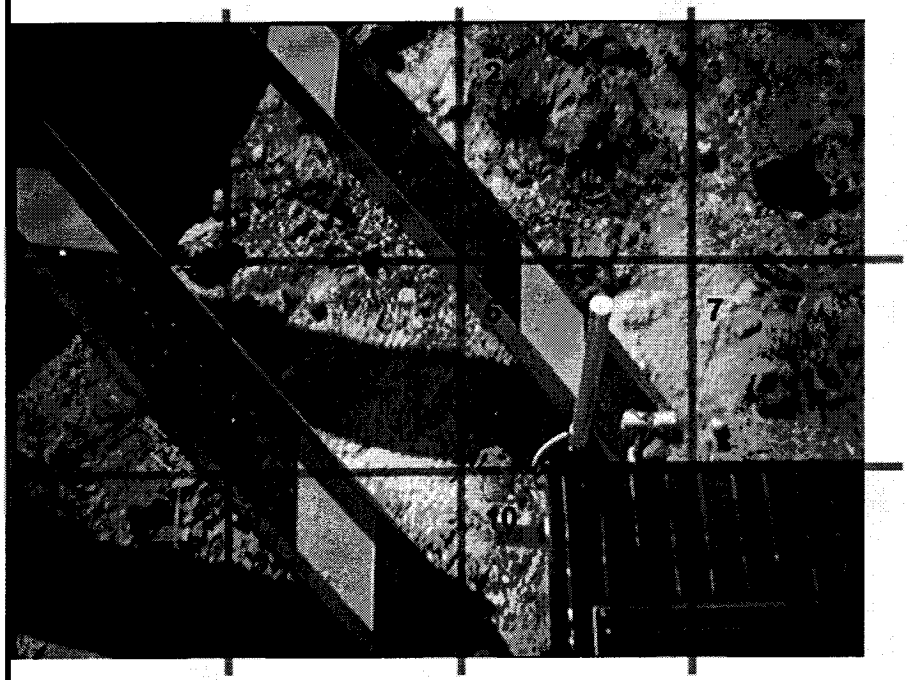
Problem

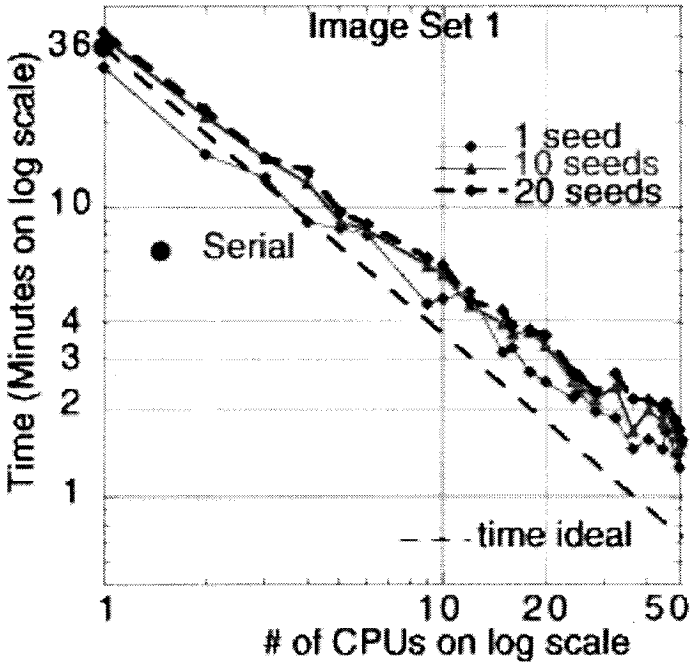
- **Need ranging data,**
  ->can deduce from relative shifts in left and right image
  ->correlation
- **Not all pixels can be correlated (parallax or terrain similarities)**
- **Computationally intensive**



JPL

Parallelization of Stereo Image Correlation





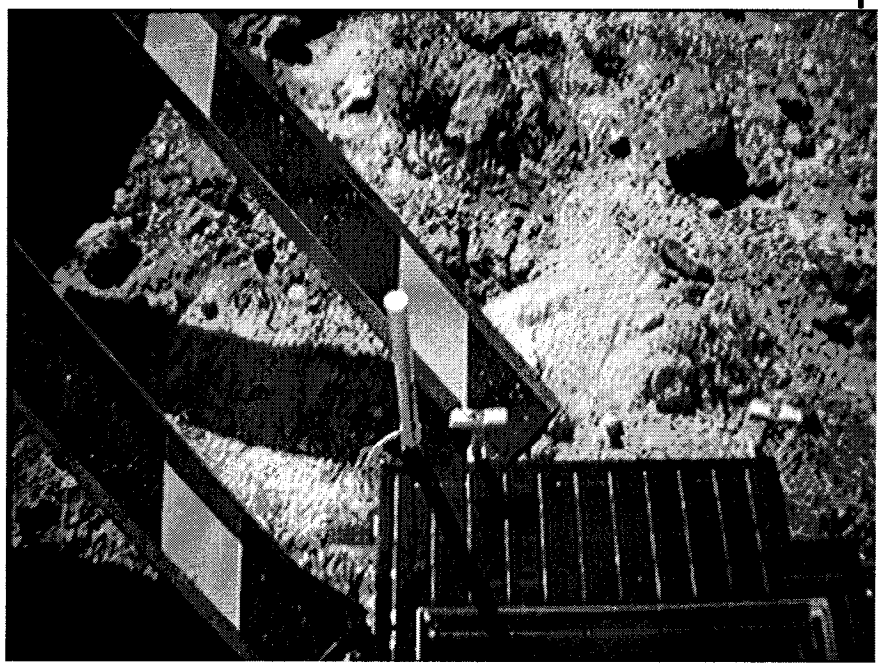
Timing Results

- Original / 450 MHz: 65 minutes
- 1 CPU / 800 MHz: 36 minutes
- 20 CPUs / 800 MHz: 3 minutes
- 50 CPUs / 800 MHz: <1.5 min.

Impact:

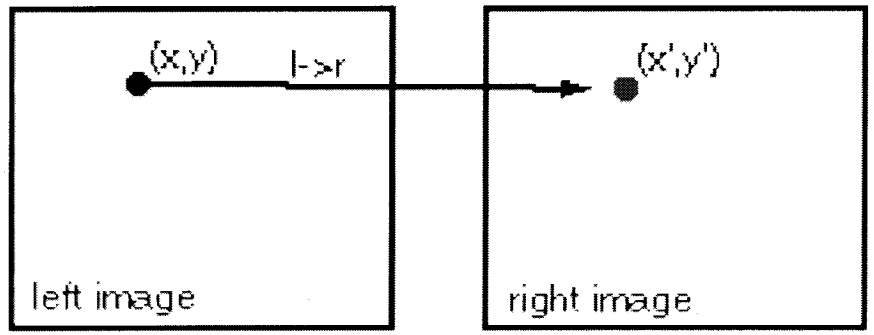
- Near real-time correlation capabilities.
- Enable QUALITY CONTROL of the correlation

Correlation Error Can Occur Unchecked!

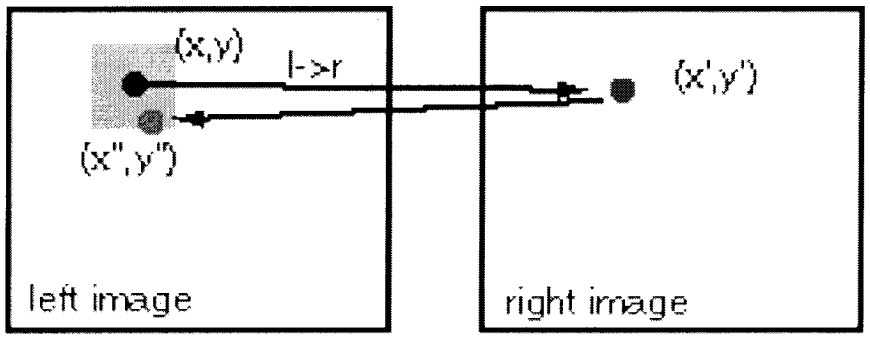


- Blue shaded pixels indicate successful correlation (from serial code)
- CPUs 31-36 work on an area that does not exist in the right image
  - segment 34 does not exist in right but was correlated -> ERROR

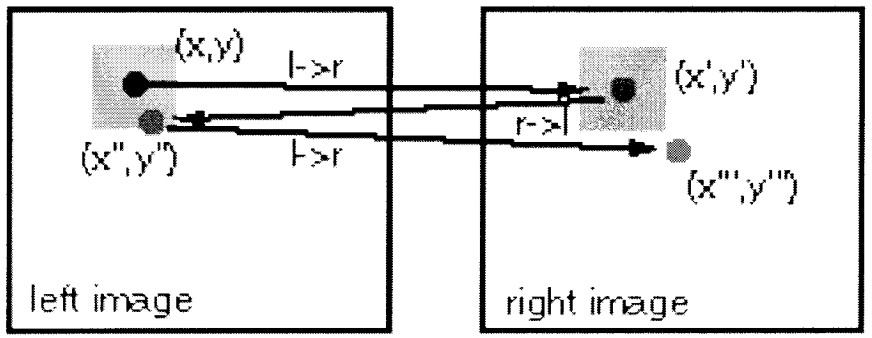
Image Correlation Quality Control Algorithm



- Desired left->right mapping

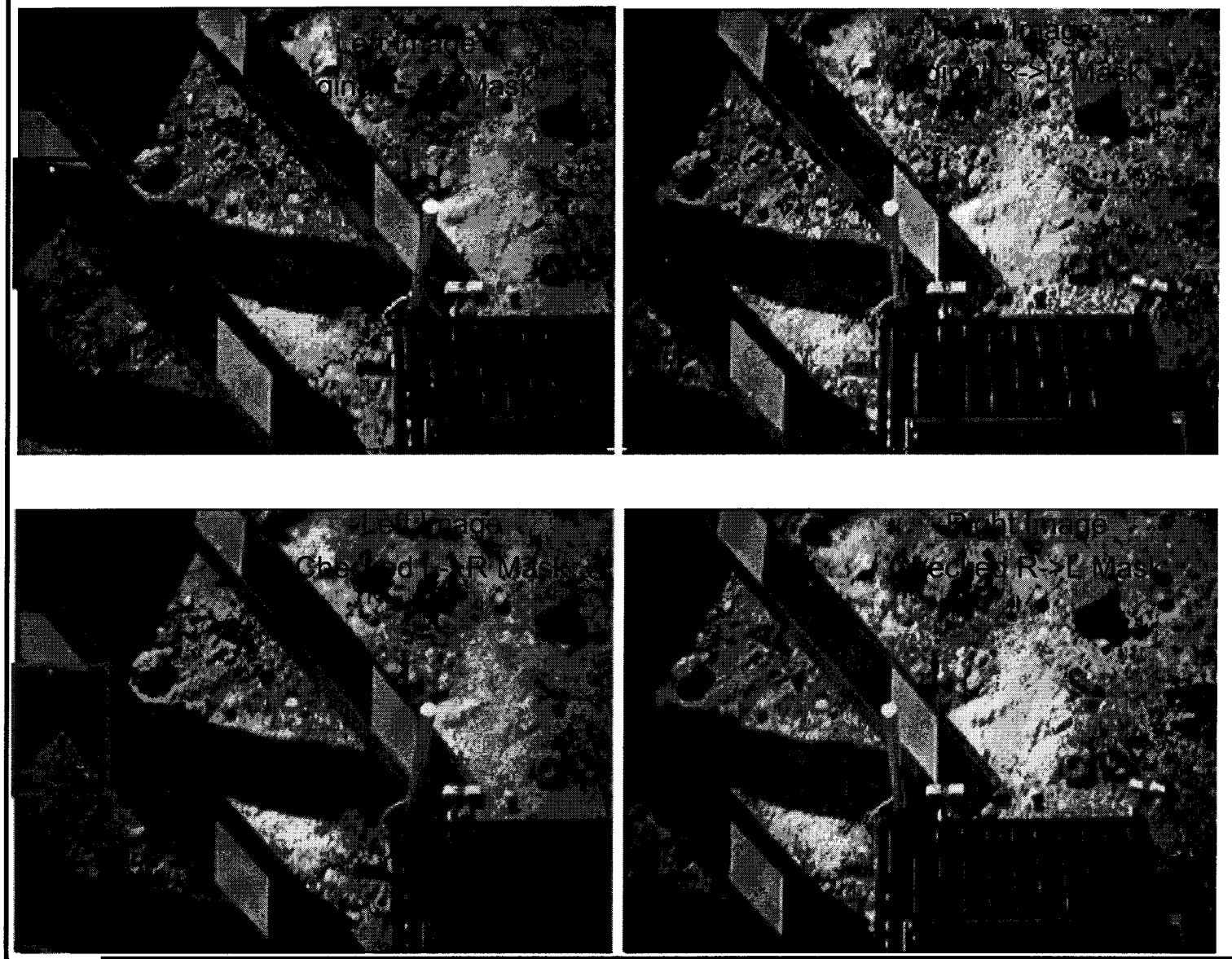


- Using a right->left mapping, can we get back to the original point?
  - Double the work load!
  - Can allow for error (yellow window)



- Self-consistent check between left->right and right->left correlation.

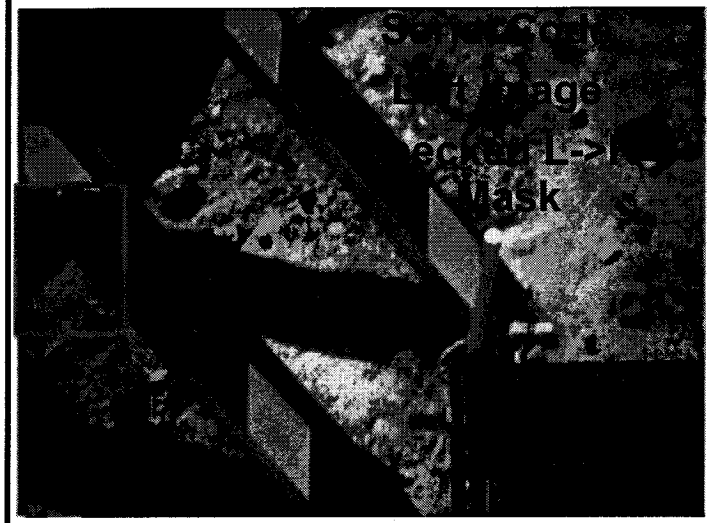
Quality Control Results

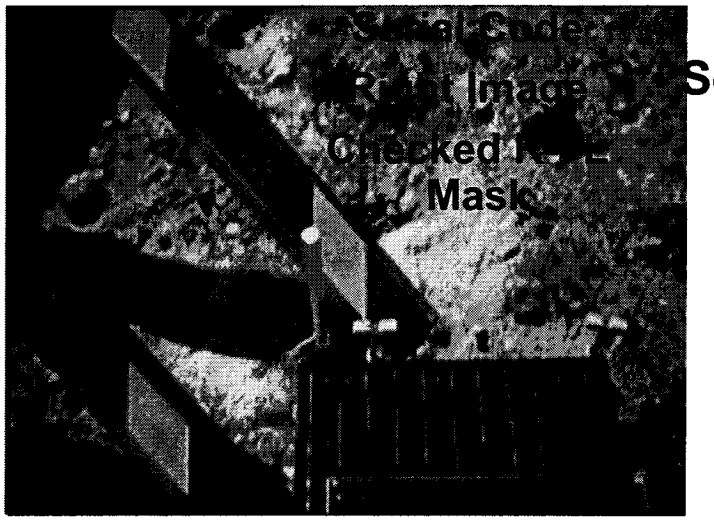


Masks before quality control

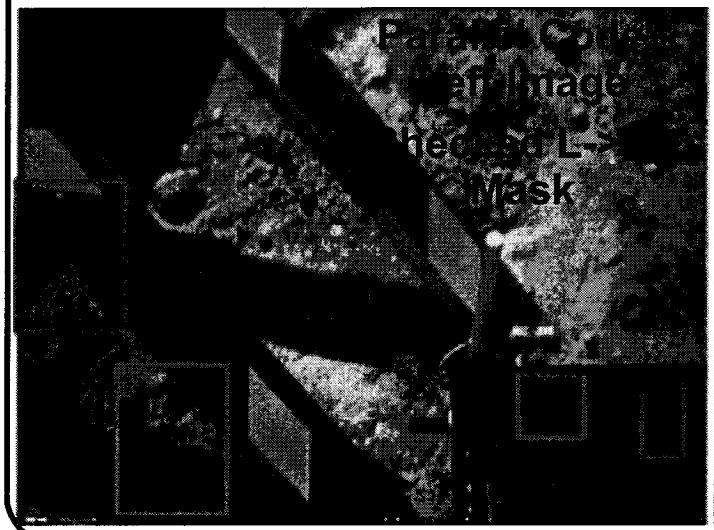
Masks after quality control

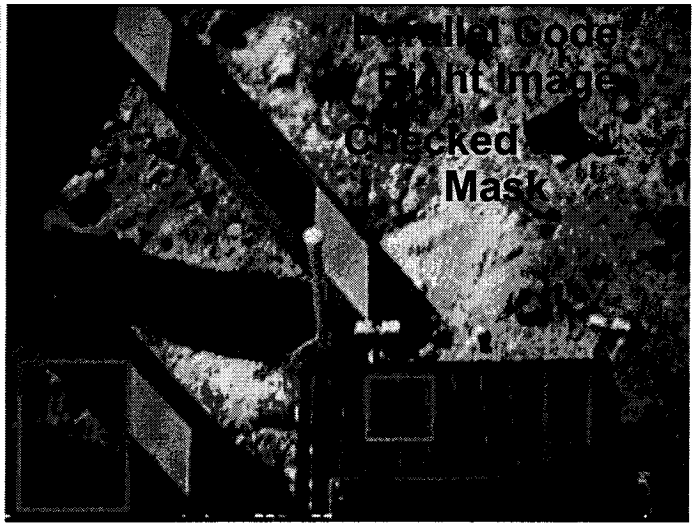
Parallel Correlation Gains/Obtains Additional Areas!





Serial results





Parallel results:

- Use many more seeds.
- Gain additional areas
- Checkerboard still a problem

Mars Image Processing on Clusters

Problem

- Current technology requires about 90min per mosaic and each correlation pair.
- Requirements are 30 minutes

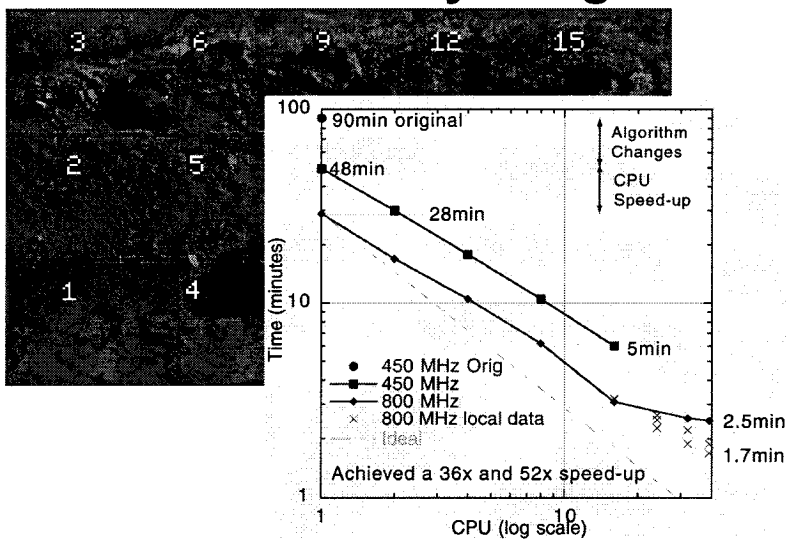
Provided Solution

- Parallelize on a commodity cluster
- Reduction mosaic time to < 3 minutes
- Reduction of correlation times to < 3 minutes
- Enable correlation quality control by running correlation twice and weeding out bad pixels

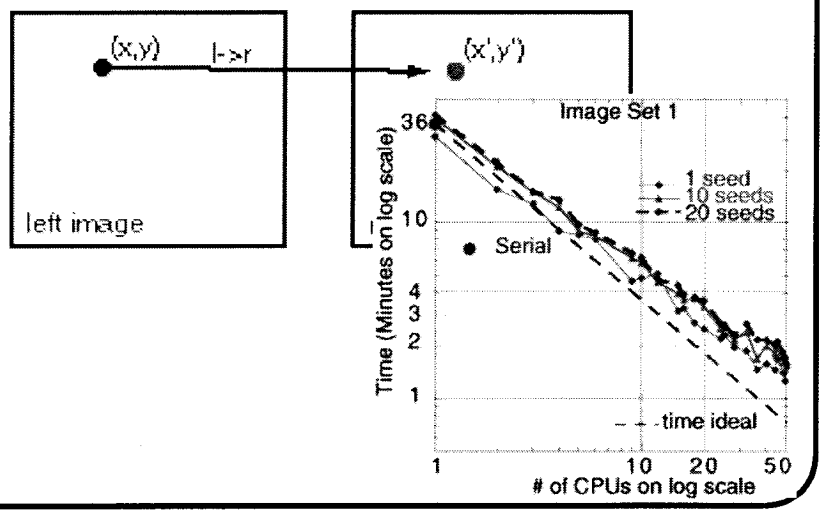
Impact

- Exceeded required time reduction
- Pixel correlation verification
- Correlator now has a forced completion, prior serial code could run indefinitely.

Mosaics from many images



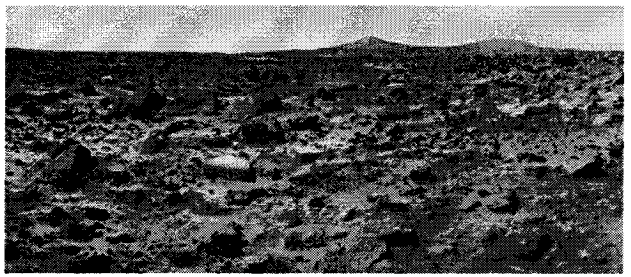
Correlation of Stereo Pairs



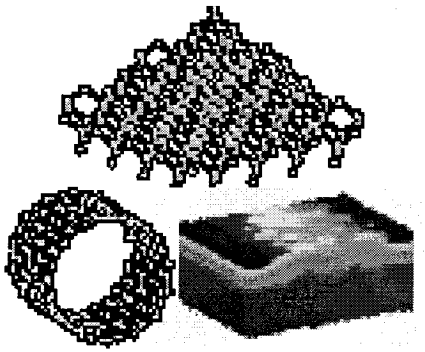
Backup Foils

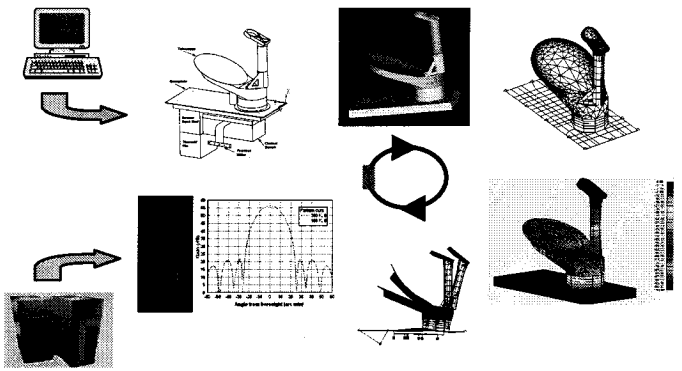
The Need for Computational Resources ...

- **Scientific Data Processing**
  - **Large data sets**
  - **Real-time results**



- **Physics-based Models**
- **High-fidelity results**
- **Large memory**
- **Long runtimes**





- **Design Environments**
- **Risk reduction**
- **Quick turn-around**

One Solution: Commodity Clusters

- Beowulf cluster-class computers leverage
  - Commodity PCs
  - Internet switching and networking
  - Open source software
- Performance proportional to # of PCs
  - Just-in-time procurement
    - Fastest cpu, RAM & disc sizing …
  - Extensible as needed
- Cost to requirements
  - Roughly $2K per node average
    - Add for memory, disc farms, cpu …

JPL

ACT

One Solution: Commodity Clusters (Cont'd)

- Three generations of experience …

Hyglac (1997)

16 Pentium Pros 200MHz
128 MB RAM per node
2 GB total
5GB Disc per node
80 GB total
100 Mb/s ethernet crossbar
Linux, MPI

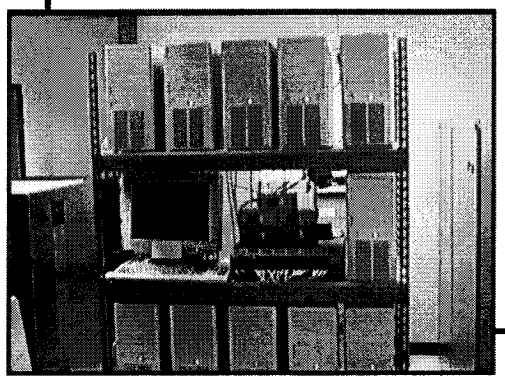
Gordon Bell Prize 1997

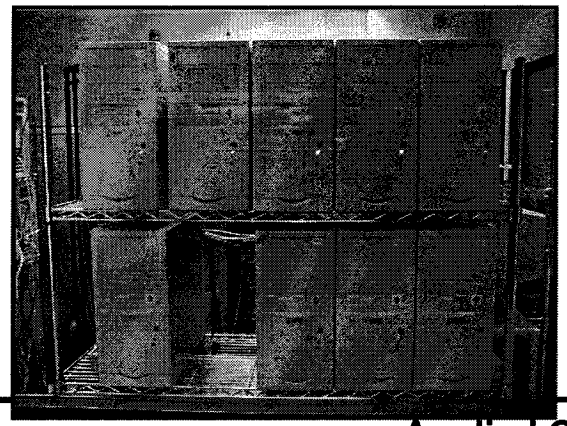
Nimrod (1999)

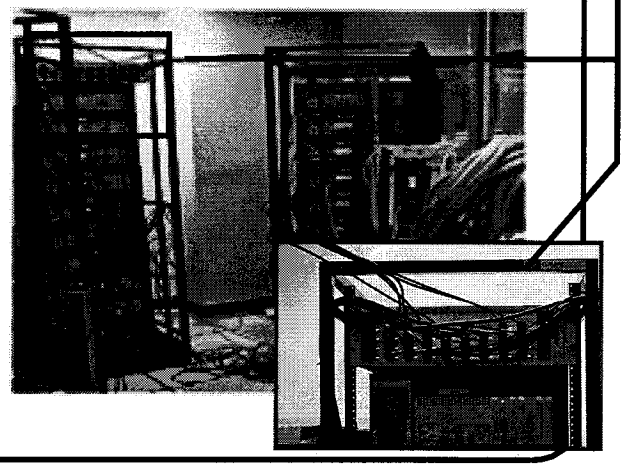
32 Pentium IIIs 450MHz
512 MB RAM per node
16 GB total
8GB Disc per node
256 GB total
100 Mb/s ethernet crossbar
Linux, MPI

Pluto (2001)

64 Pentium IIIs 800MHz
dual CPUs
2 GB RAM per node
64 GB total
10 GB Disc per node
320 GB total
2 Gb/s Myricom crossbar
Linux, MPI



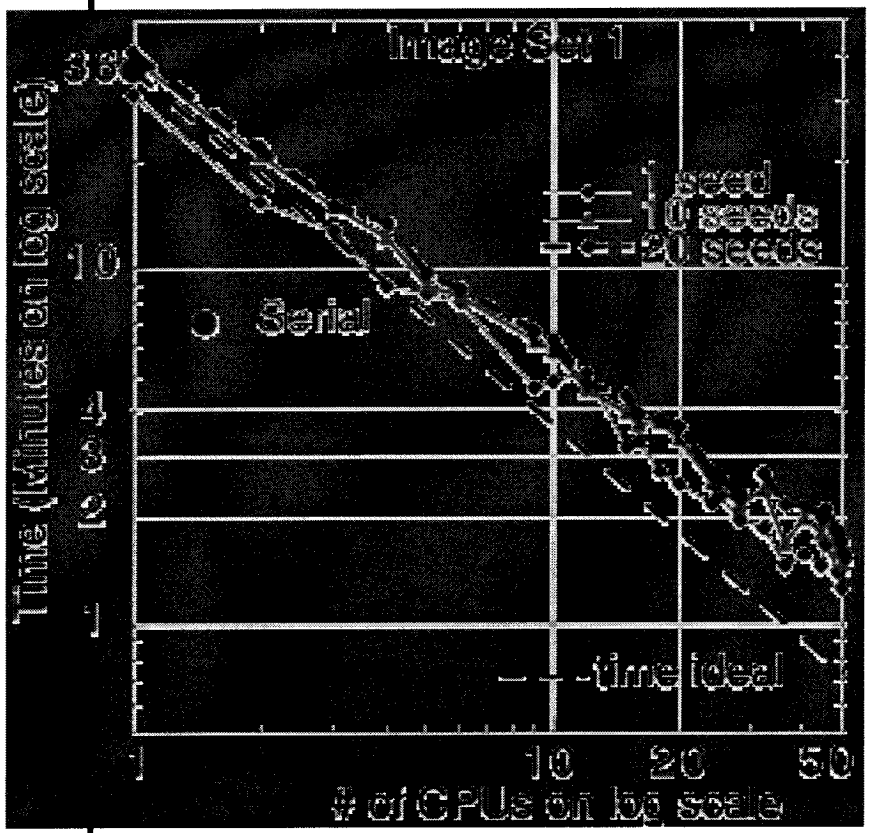




But Software (Applications) is still the Issue …

- Cost and availability of hardware is enabling
  - BUT …
- Applications need to utilize hardware
  - Task Farms: schedule jobs to execute as needed on processors
    - Data processing, computation + visualization …
  - Simple Parallelization: execute same code with varying parameters on processors
    - Optimization, parameterization sweeps …
  - Complete Parallelization: distribute memory and computation
    - Physics-based high fidelity modeling

Application : Image Correlation (Cont'd)



Timing Results

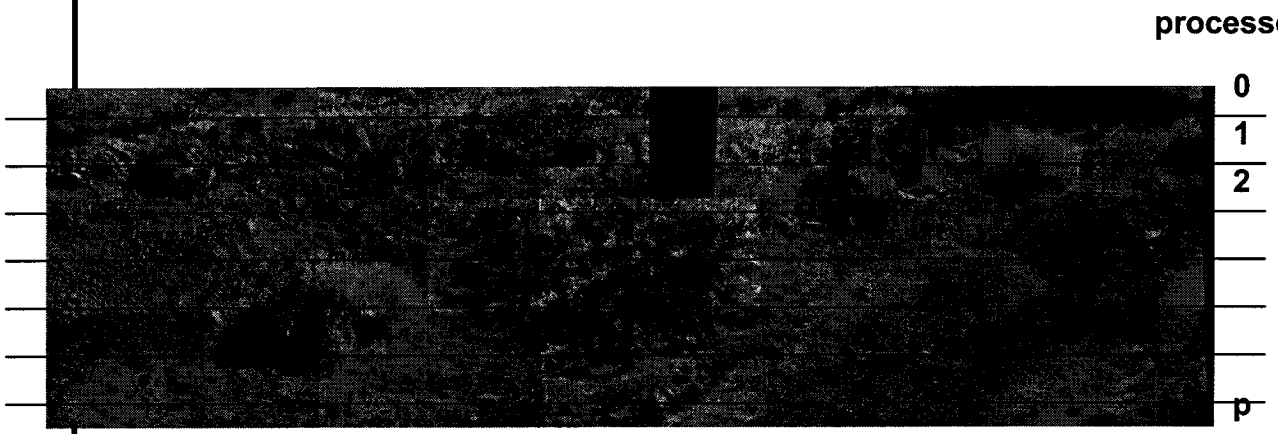
- Original / 450 MHz: 65 minutes
- 1 CPU / 800 MHz: 36 minutes
- 20 CPUs / 800 MHz: 3 minutes
- 50 CPUs / 800 MHz: <1.5 min.
- **Where does the non-ideality come from?**

Quality of the correlation:

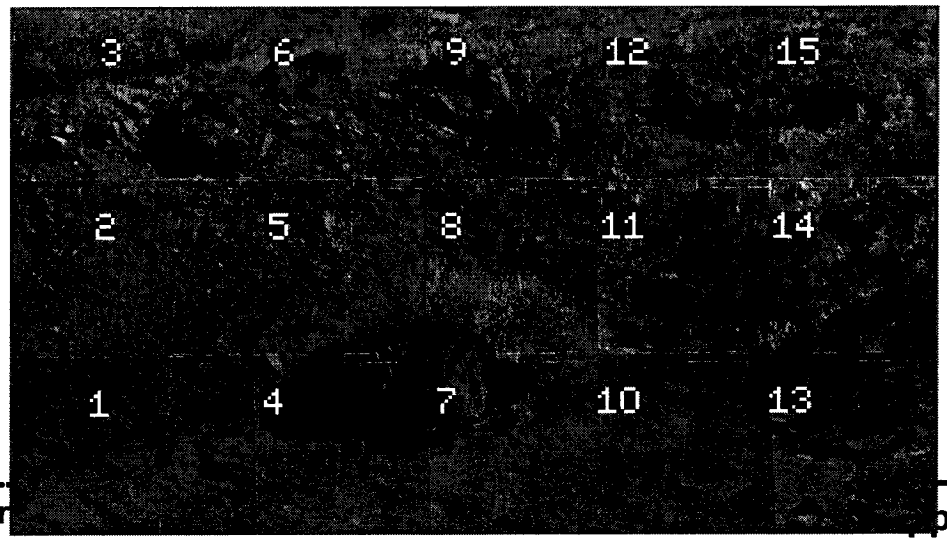
- Number of correlated pixels: varies with the number c CPUs, but it is approximately the same
- **How good are the correlation pixels?**
- Do the seams between correlation areas matter?

Application : Mars Mosaic Production (Cont'd)

- Images from FIDO field test on 4/30/01-5/2/01



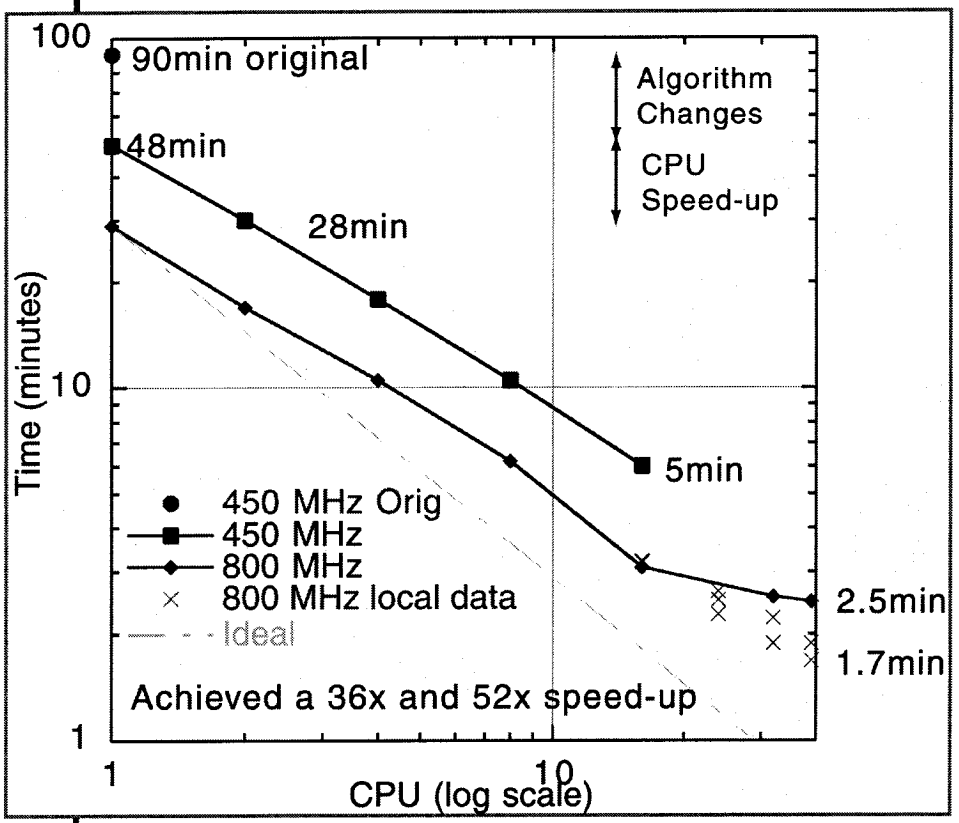
Divide panorama (output) by lines and distribute to processors



For each panorama mosaic line and sample, find pixels in corresponding (input) image

JPL

Application : Mars Mosaic Production (Cont'd)

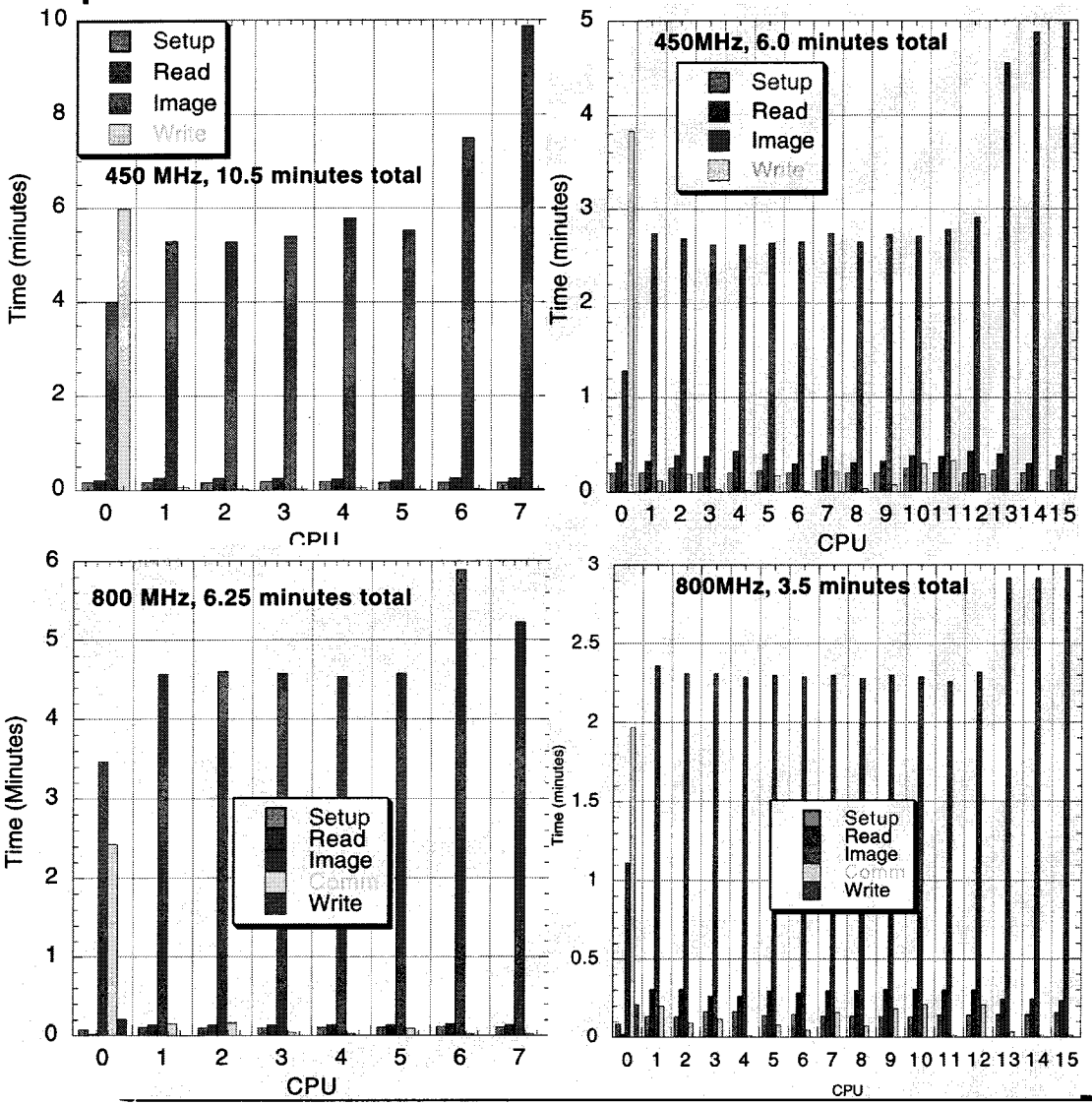


- Starting from 123 individual 312kB images (38MB).
- Final mosaic contains 3365 lines and 19975 samples resulting in a 134MB graphics file (with dark areas).

- Starting point 450MHz: 90 minutes
- Algorithm Changes -
  Storage of all images: 48 minutes
- 16 CPUs / 450MHz: 5 minutes
- 40 CPUs / 800MHz: 2.5 minutes
- with local data:
  40 CPUs / 800MHz: 1.7 minutes /

- Deviation from ideal scaling:
  - Load balancing
  - Data staging from the front end disk

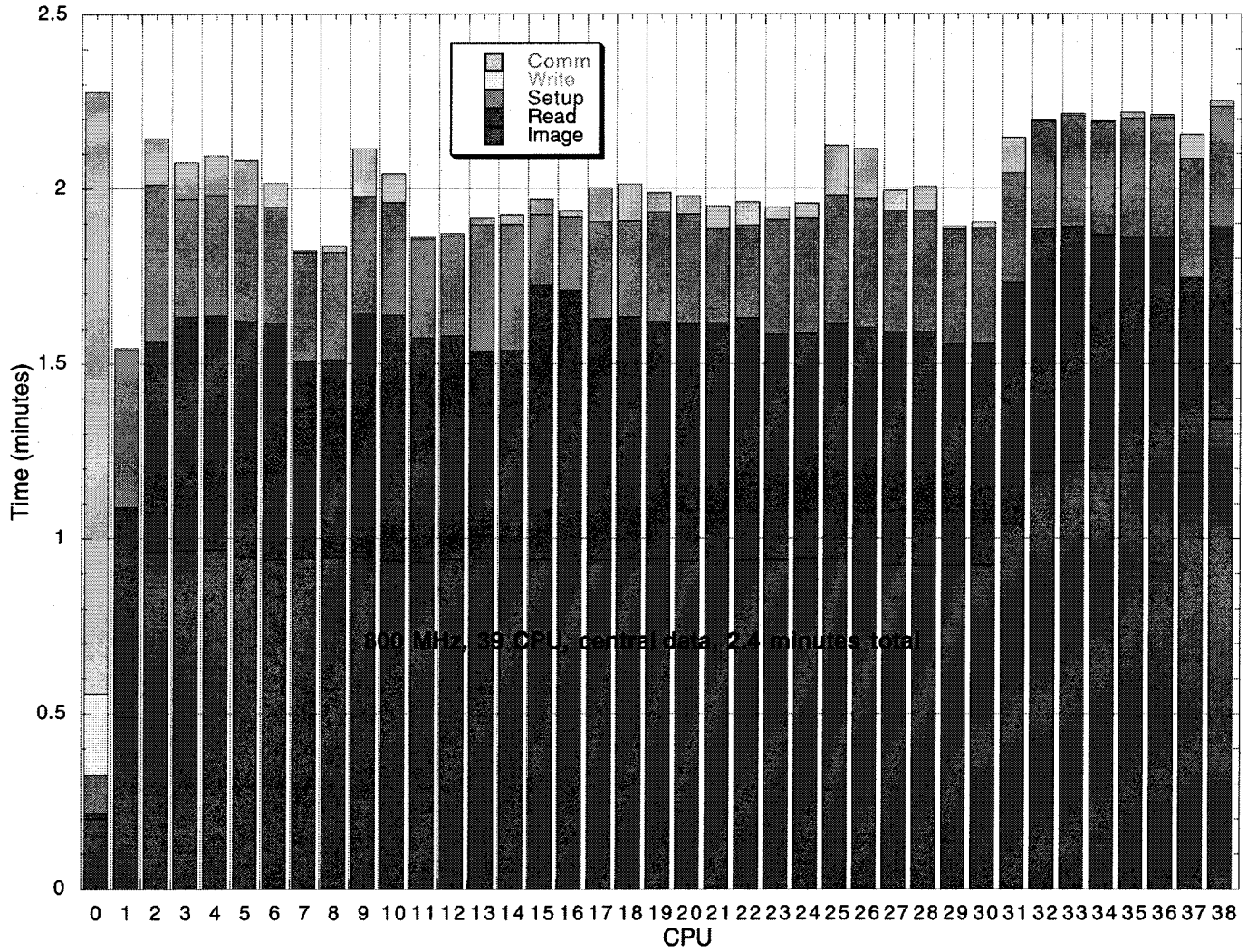
Load Balance Analysis: 8 and 16 CPUs



Examples on 8 and 16 CPU's:

- **Good:**
  - **Most of the time is spent on the actual image processing!**
  - **Set-up and read times are well balanced**
- **Bad:**
  - **Actual image processing is not well balanced!**
    -especially for 450MHz
    -less pronounced 800MHz

Load Balance Analysis: 39 CPUs

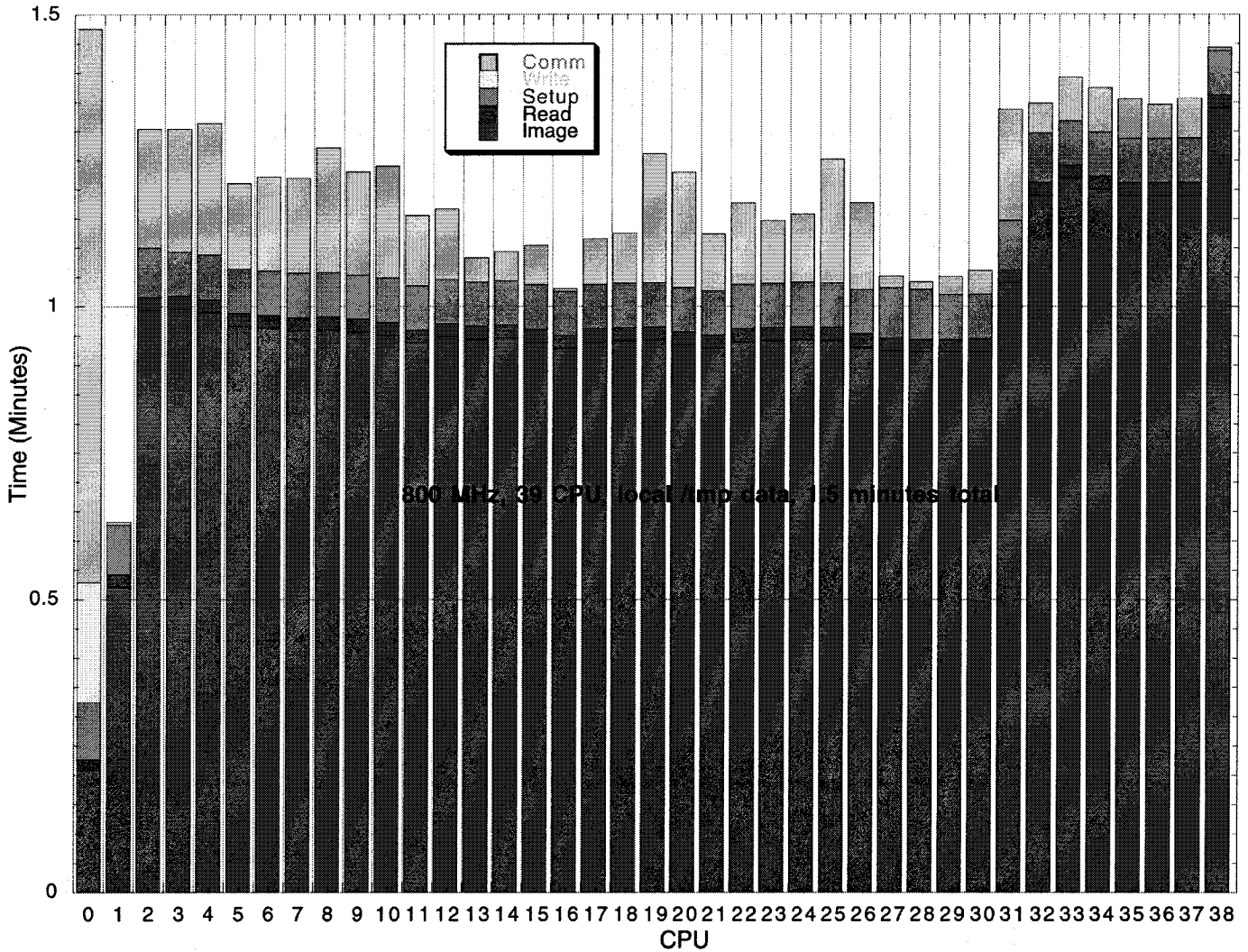


Initial file reading becomes a major total time contributor.

JPL

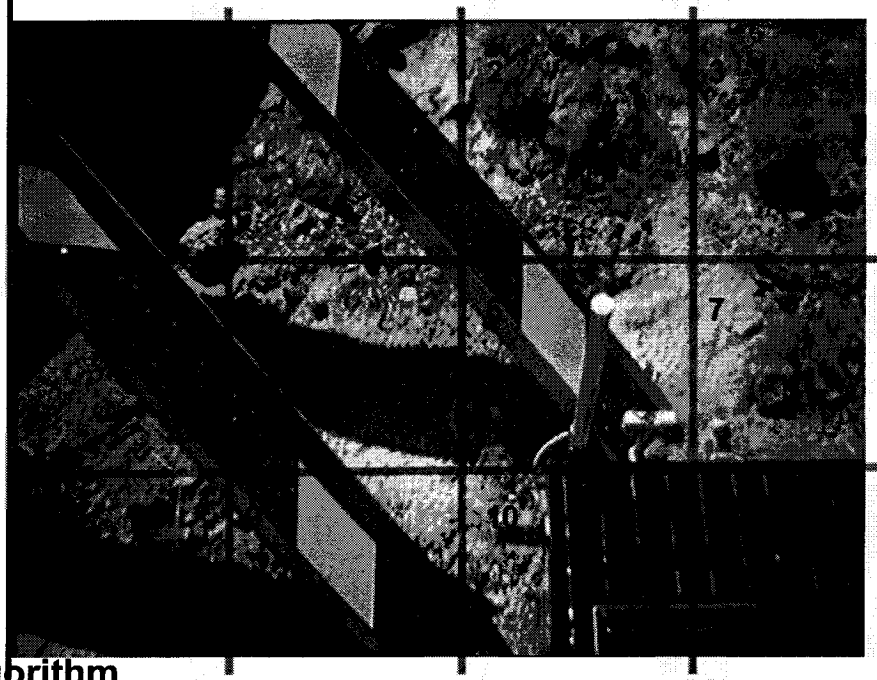
ACT

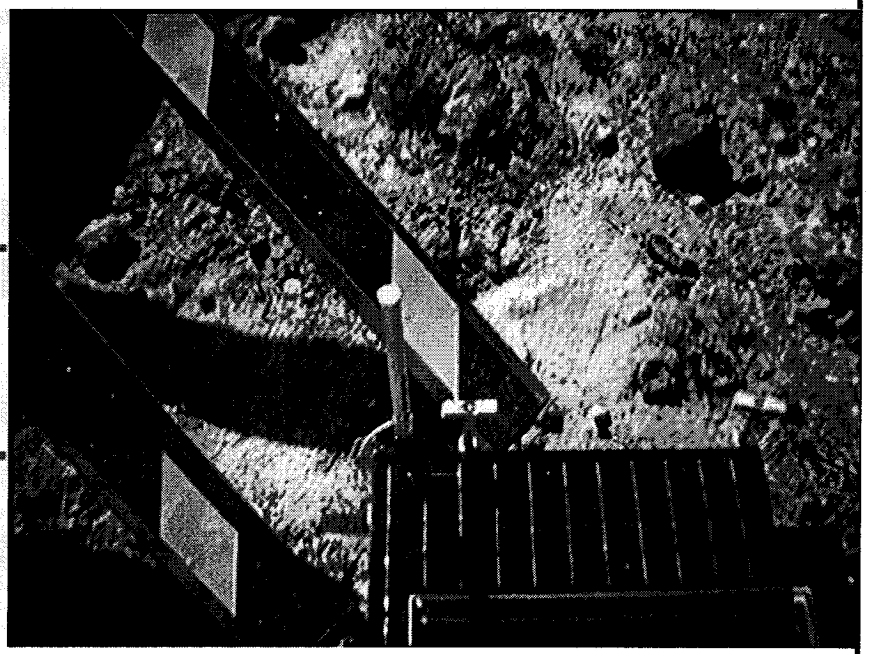
Load Balance Analysis: 39 CPUs local /tmp data



Local data reduces the total time by about a minute (50%)
Image processing load balance could improve performance by ~20-40%.

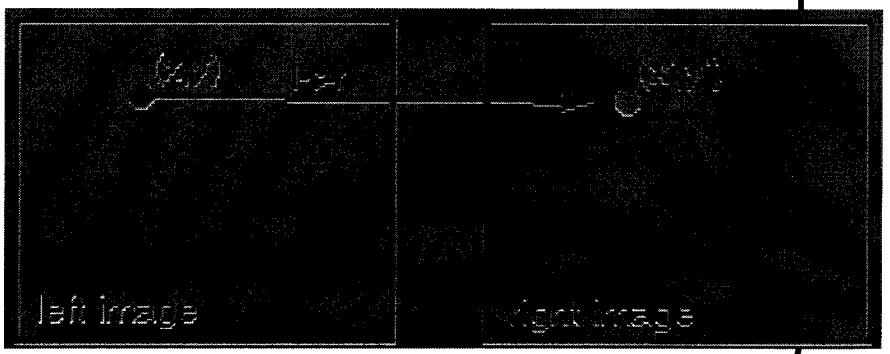
Application: Left&Right Stereo Image Correlation



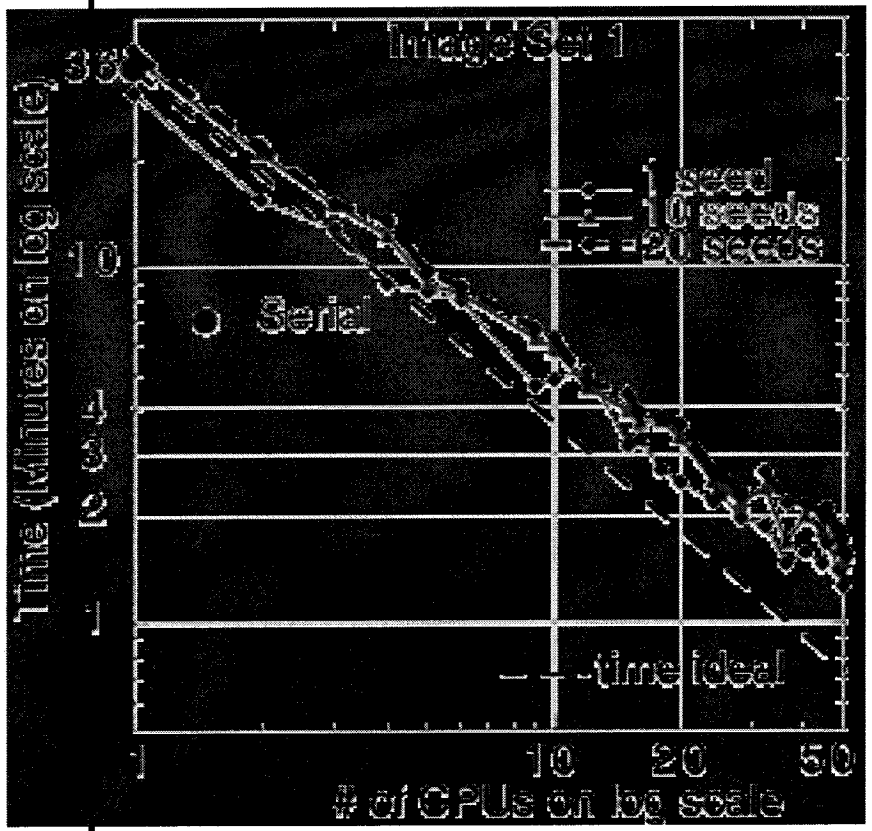


Algorithm

- Pixel in left image of stereo-pair is correlated with pixel in right image.
- Not all pixels can be correlated (parallax or terrain similarities)
- Divide work in left image; processor holds both images and output correlation masks and disparity data



Application : Image Correlation (Cont'd)



Timing Results

- Original / 450 MHz: 65 minutes
- 1 CPU / 800 MHz: 36 minutes
- 20 CPUs / 800 MHz: 3 minutes
- 50 CPUs / 800 MHz: <1.5 min.
- **Where does the non-ideality come from?**

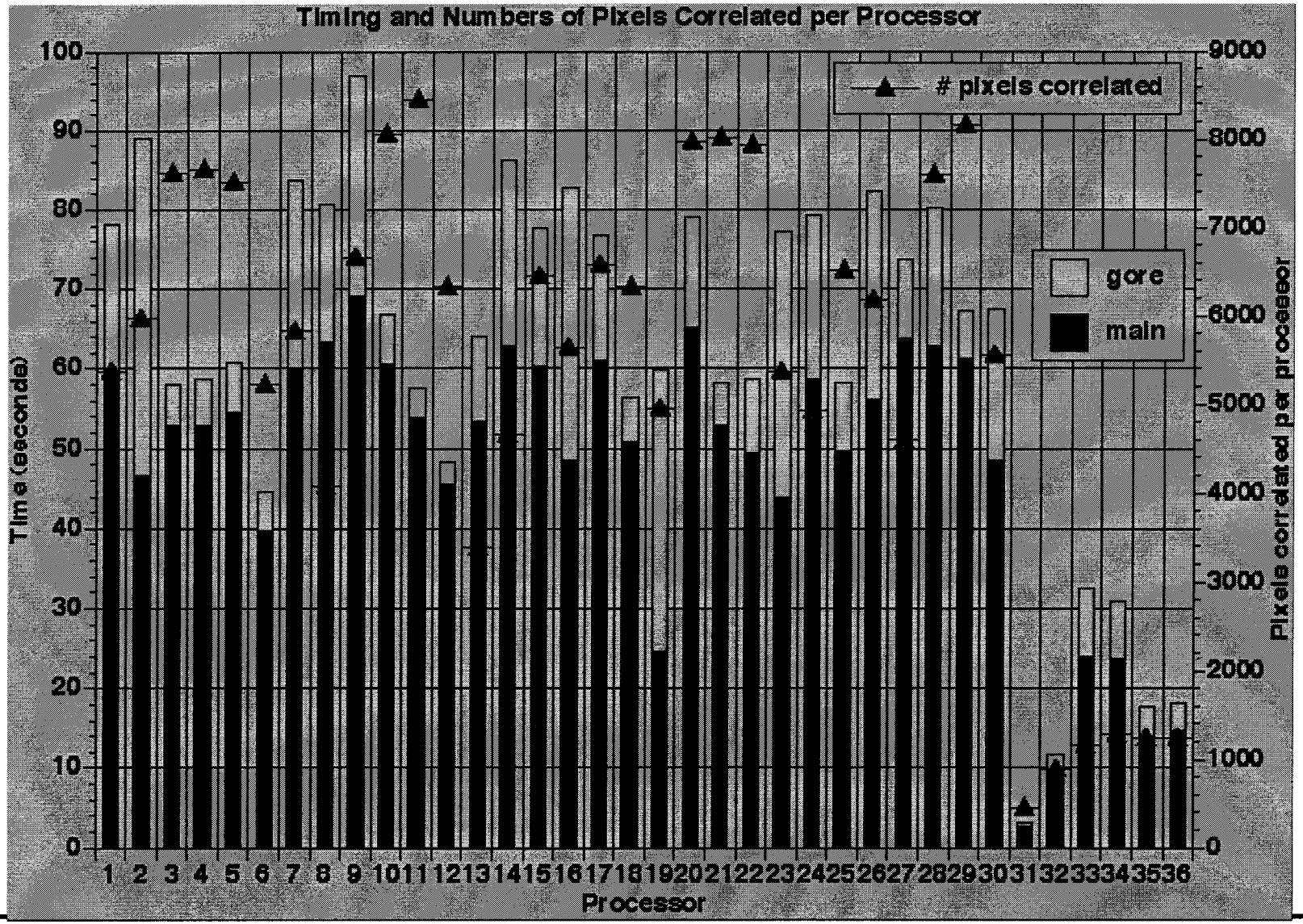
Quality of the correlation:

- Number of correlated pixels: varies with the number c CPUs, but it is approximately the same
- **How good are the correlation pixels?**
- Do the seams between correlation areas matter?

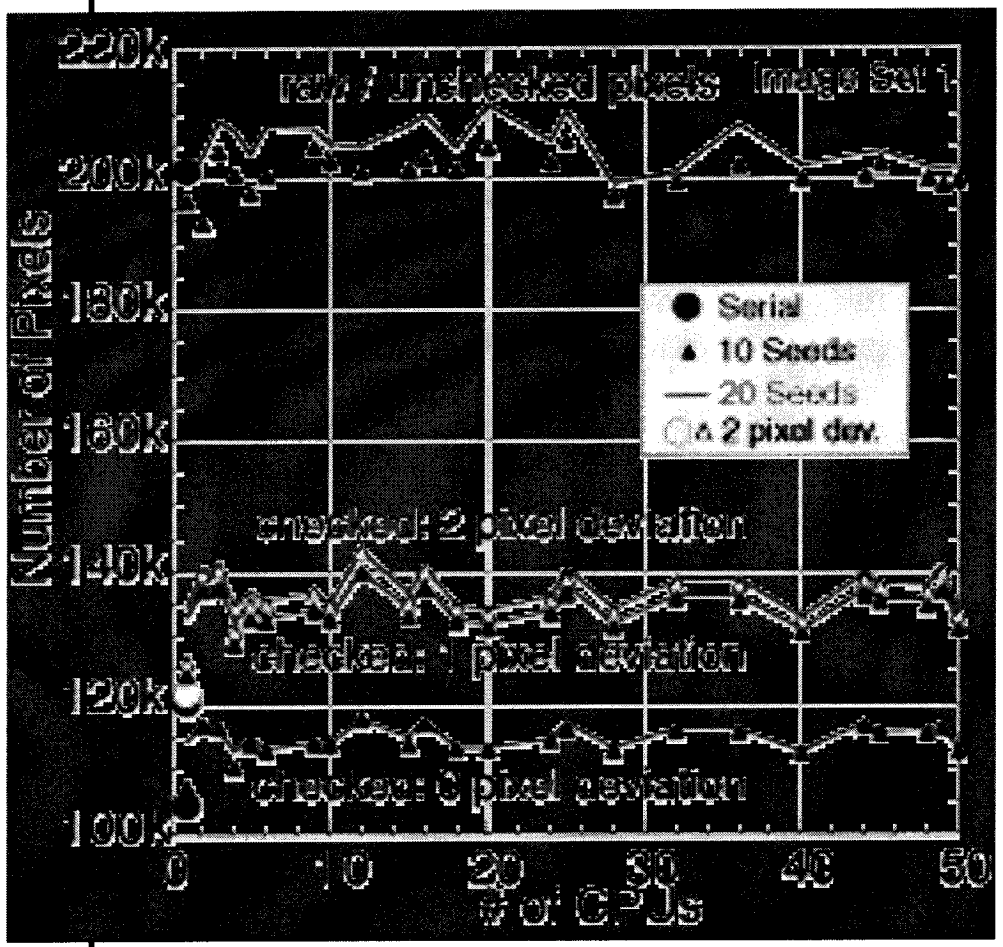
JPL

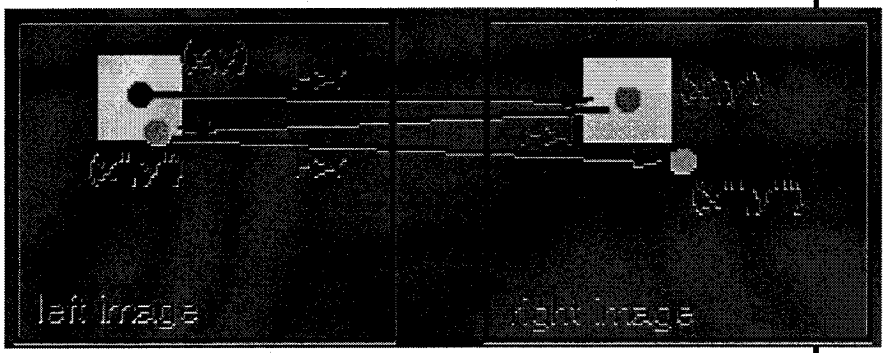
ACT

plication: Image Correlation - Load Balance Analys



Application : Image Correlation - Quality Control



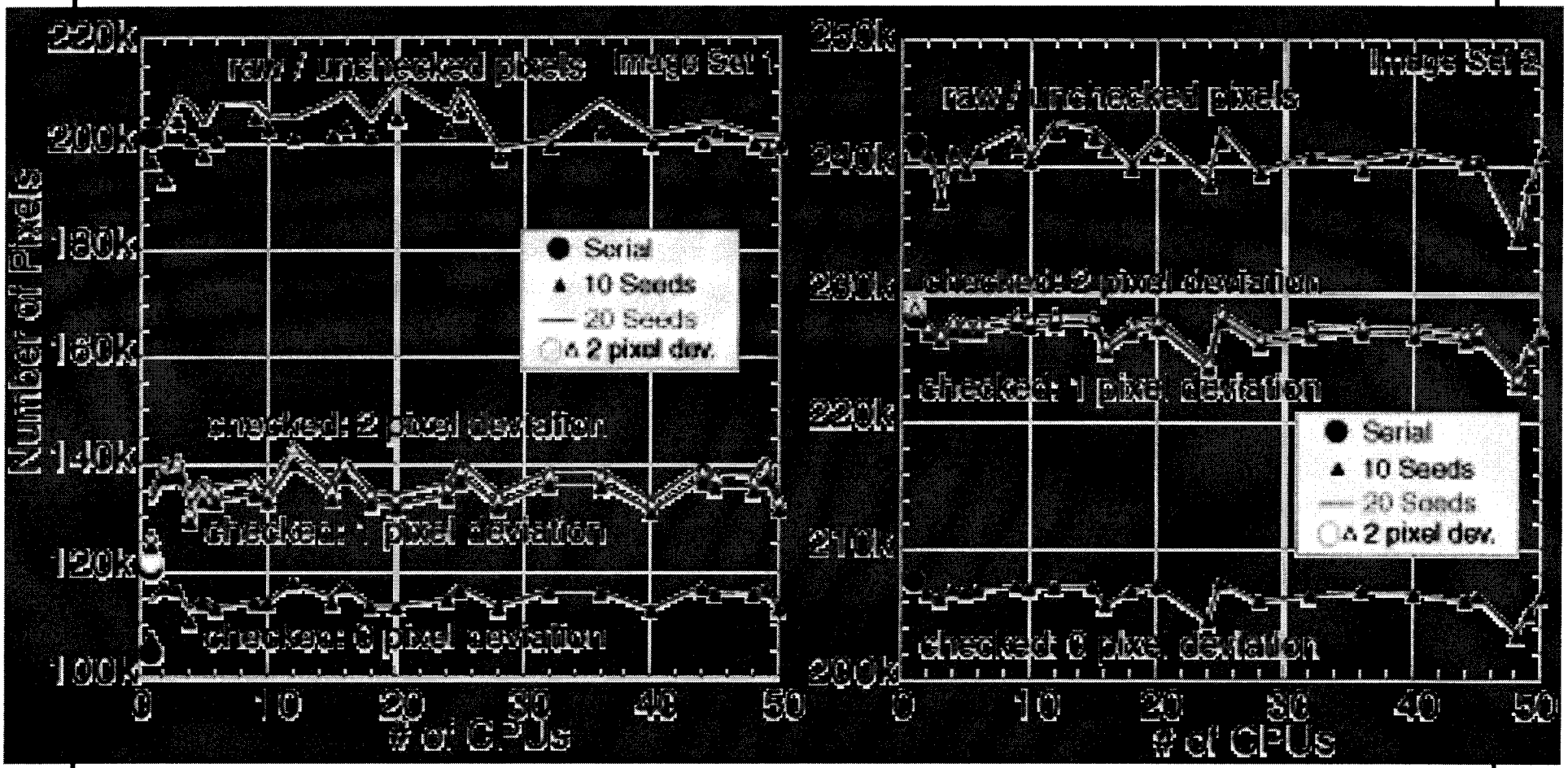


- 0 pixel error: too restrictive due to 1/2 pixel round-off
- 1-2 pixel error acceptable
- Parallel code just as good as serial code or better than serial code

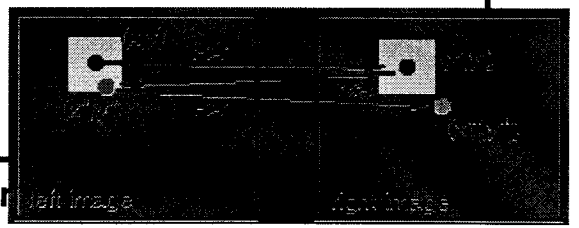
JPL

ACT

Application : Image Correlation - Quality Control



- 0 pixel error: too restrictive due to 1/2 pixel round-off
- 1-2 pixel error acceptable
- Parallel code just as good as or better than serial code



Conclusions

- **Showed near-realtime image processing application for mars rover operations on beowulf systems**
- **Mars Mosaic Generation**
  - **Reduced original compute time from 90 minutes to ~2 minutes**
- **Mars Stereo Image Pair Correlation**
  - **Reduced original from 65 minutes to less than 2 minutes**
  - **Developed a correlation quality measurement tool**
    - Discard bad correlation pixels
    - Parallel code is just as good as serial code
    - This enabled us to evaluate simplified correlation algorithms Reduce computing time to **20 seconds** at 10% pixel loss (not shown)

Applications that Utilize the Clusters ...

- **Science data processing**
  - **MODTRAN radiative transfer code**
  - **MARSMAP mosaic generation**
  - **CORRELATION of rover stereo image pairs**
- **Physics-based modeling**
  - **Outer-planet atmospheric convection model**
  - **MATPAR: parallel cluster interface to MATLAB**
  - **Antenna (reflector and patch) models**
  - **PYRAMID: parallel adaptive mesh refinement library**
  - **Infrared filter design and optimization**
  - **Nanoelectronic Modeling**
- **Design environments**
  - **MODTOOL:** Millimeter-wave Optics Design Tool
  - Genetic Algorithm design and optimization; risk reduction simulation